AutoLRS: Automatic Learning-Rate Schedule by Bayesian Optimization on the Fly

Yuchen Jin, Tianyi Zhou, Liangyu Zhao, Yibo Zhu,
Chuanxiong Guo, Marco Canini, Arvind Krishnamurthy

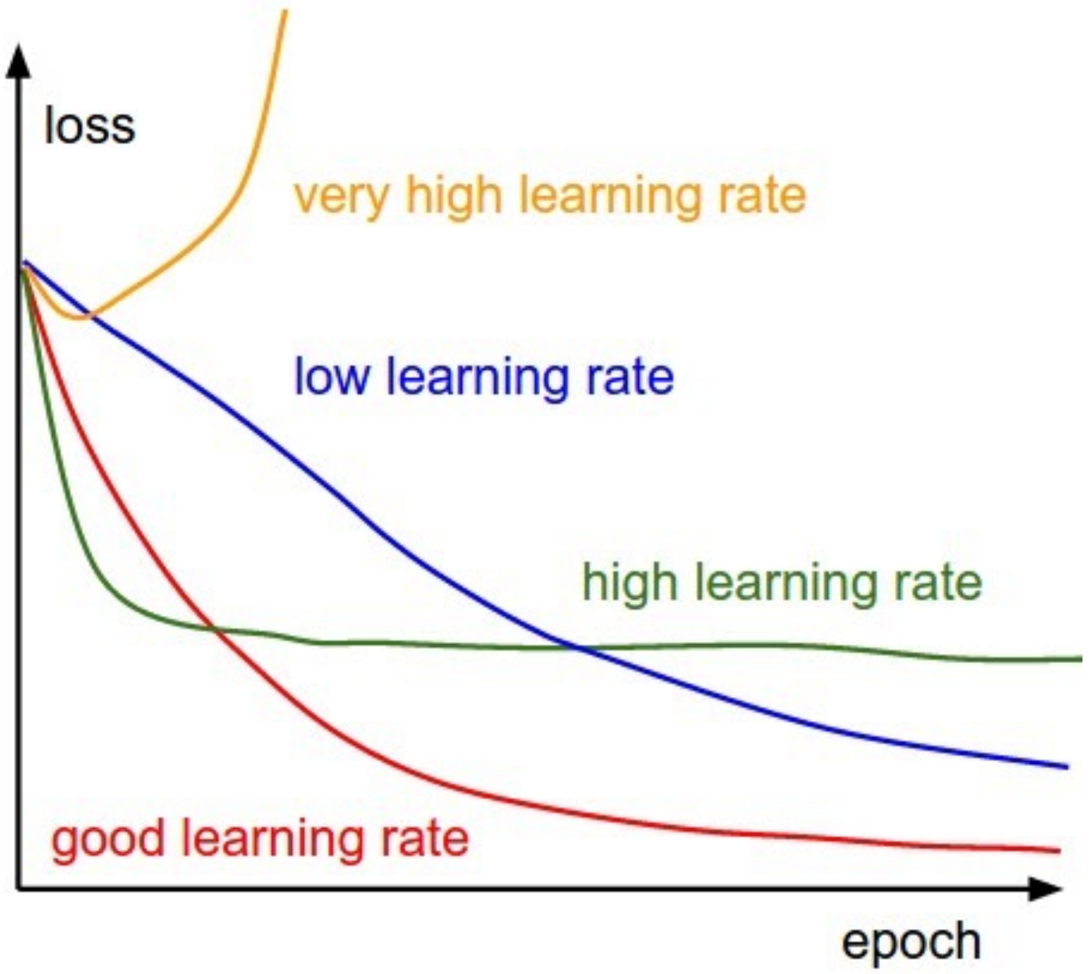
UNIVERSITY of WASHINGTON

ByteDance

KAUST

Learning rate (LR)

- Learning rate is a parameter that determines the step size at each iteration of the optimization problem.
- The success of training DNNs largely depends on the LR schedule.



[Stanford CS231n]

Tuning the learning rate (LR) schedule is non-trivial

Widely-used tuning strategies:

- Pre-defined LR schedules
  - Limited number of choices, e.g., step decay & cosine decay
- Optimization methods with adaptive LR (such as Adam and AdaDelta)
  - Still require a global learning rate schedule: Adam's default LR performs poorly in training BERT and Transformer

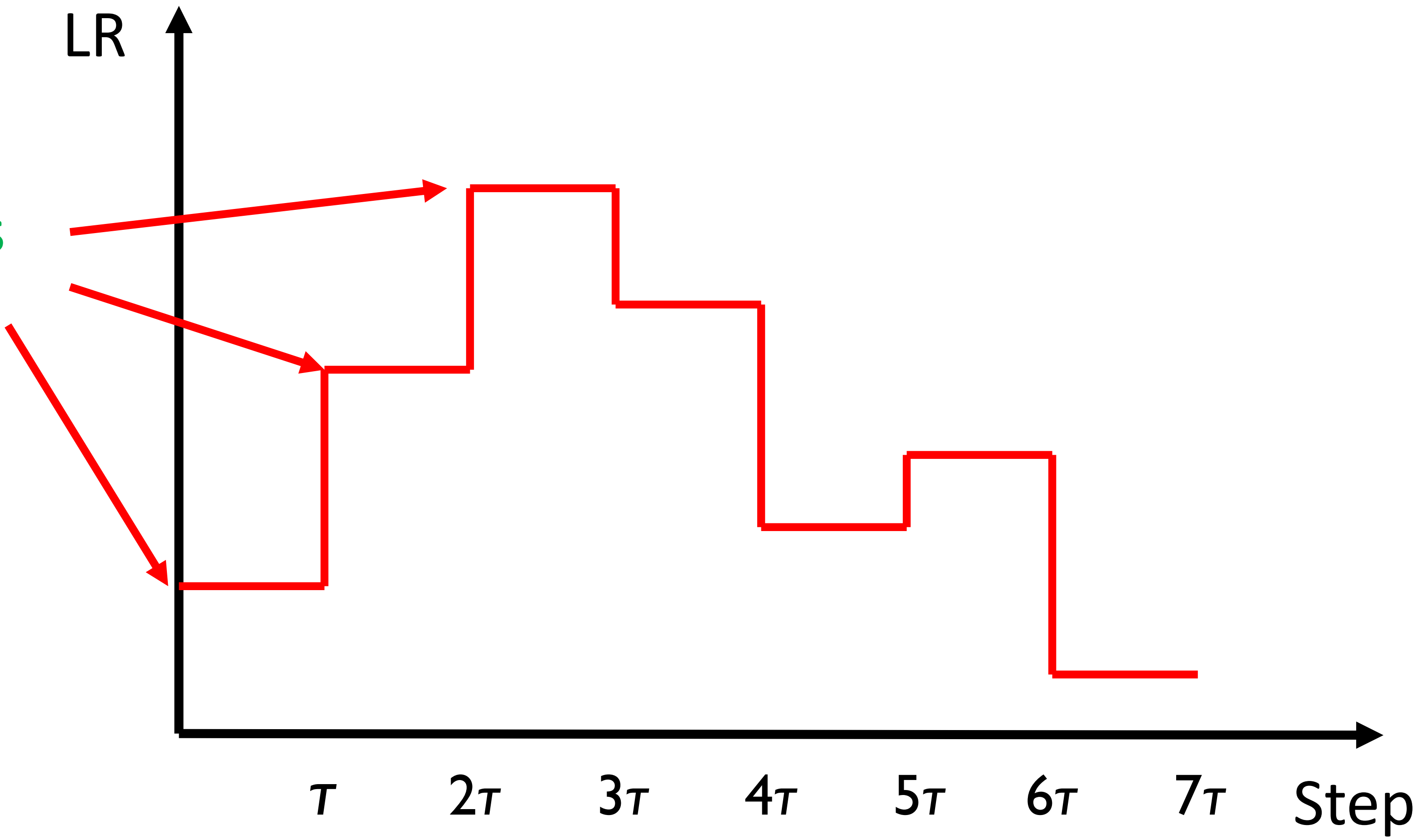
Both strategies introduce new hyper-parameters that have to be tuned separately for different tasks, datasets, and batch sizes.

Can we automatically tune the LR over the course of training without human involvement?

AutoLRS

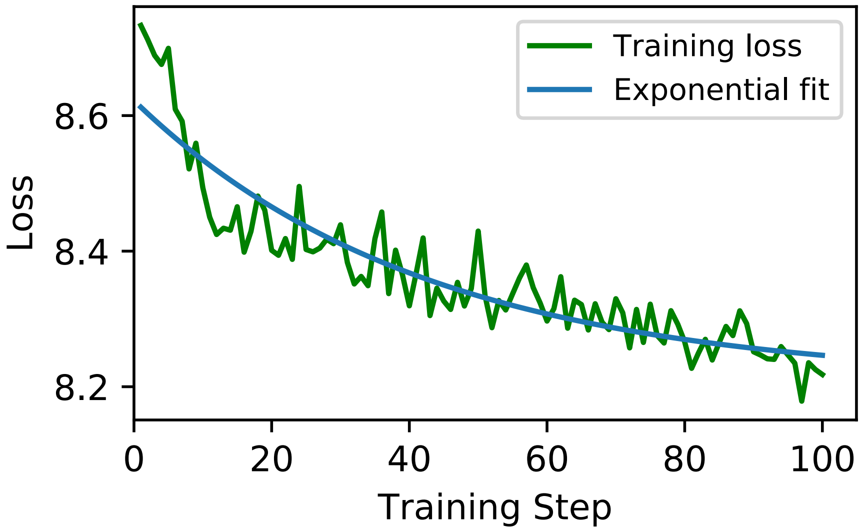
Coarse-grained approach: determining a constant LR for every τ steps ⇒ "training stage"

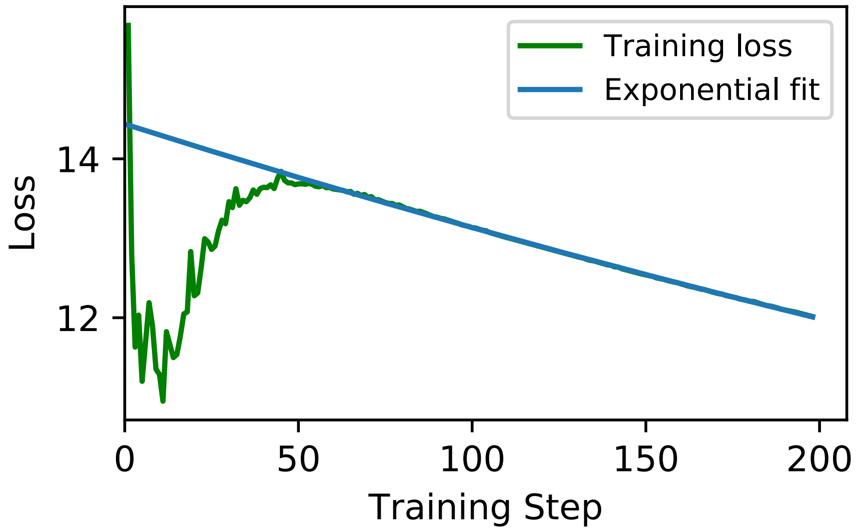
Which LR can minimize the validation loss for the next *stage*?



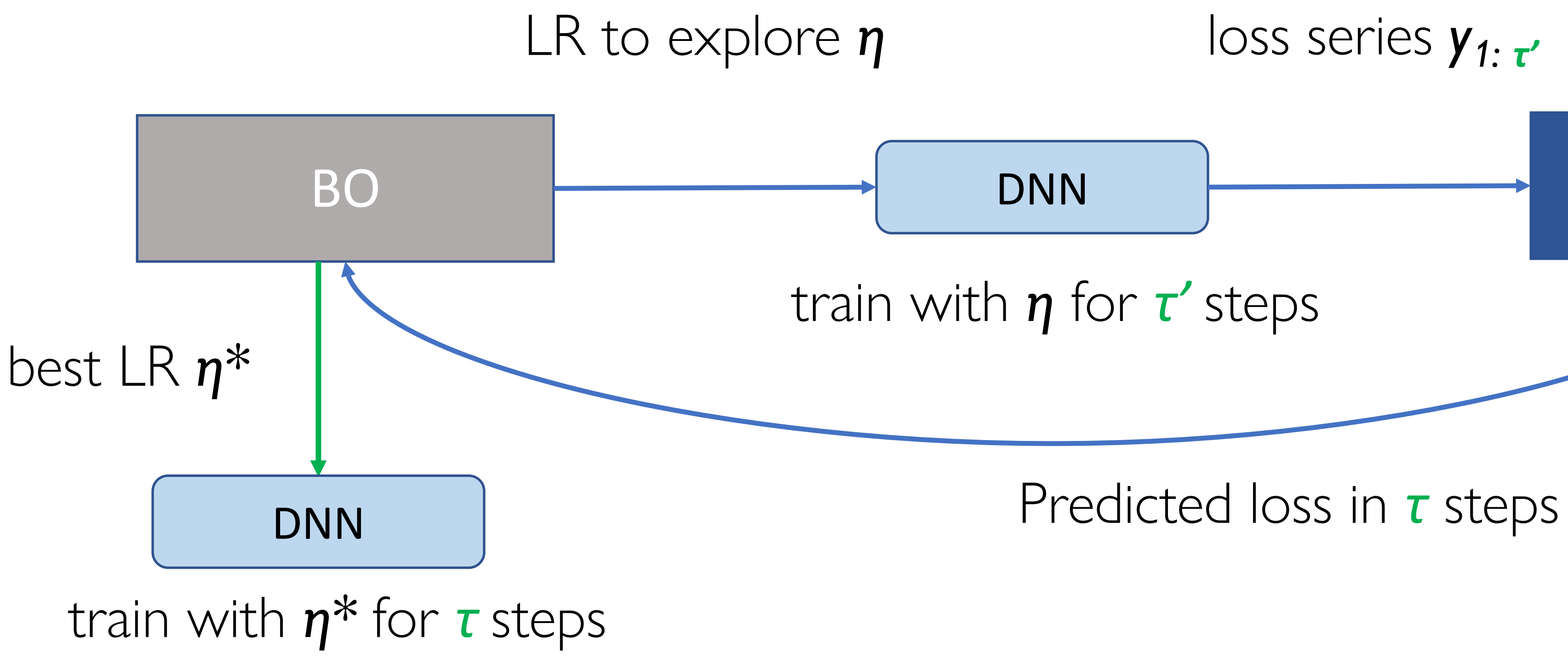
AutoLRS

- ## Bayesian optimization (BO)
  - Treat the validation loss w.r.t. LR as a black-box function.
  - BO would require τ training steps to measure the validation loss associated with every LR η it explores. → Computationally expensive
- ## Exponential forecasting model
  - For each LR η that BO explores, we only apply it for $\boldsymbol{\tau' \ll \tau}$ steps and use the validation loss observed in the τ' steps to train a time-series forecasting model.





Search for the LR at the beginning of each *stage*



- Each LR evaluation during BO starts at the same model parameter checkpoint
- $\tau' = \tau/10$; BO explores 10 LRs in each stage
  → steps spent to find the LR
  = steps spent on training the model with the identified LR.

Experiments

- **Models:** ResNet-50, Transformer, BERT Pre-training

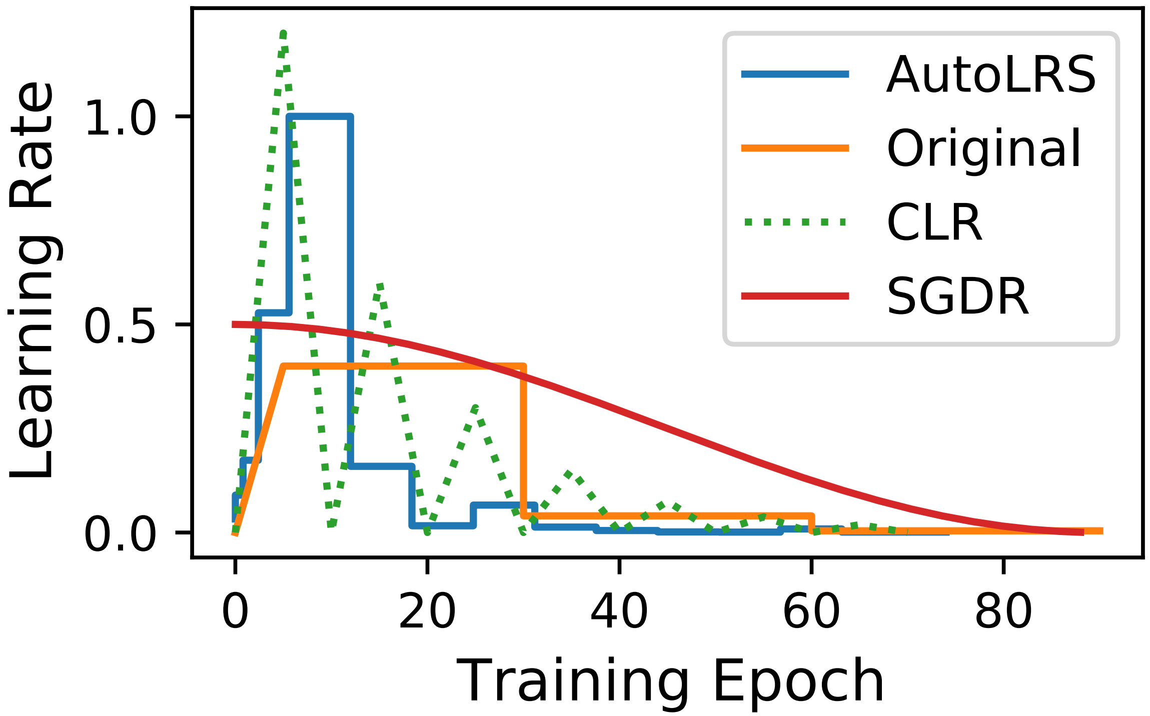
- **Baselines:**
  - LR schedule adopted in each model's original paper
  - Highly hand-tuned Cyclical Learning Rate (*CLR*) [1]
  - Highly hand-tuned Stochastic Gradient Descent with Warm Restarts (*SGDR*) [2]

[1] Leslie N Smith. Cyclical learning rates for training neural networks. WACV'17.
[2] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. ICLR'17.

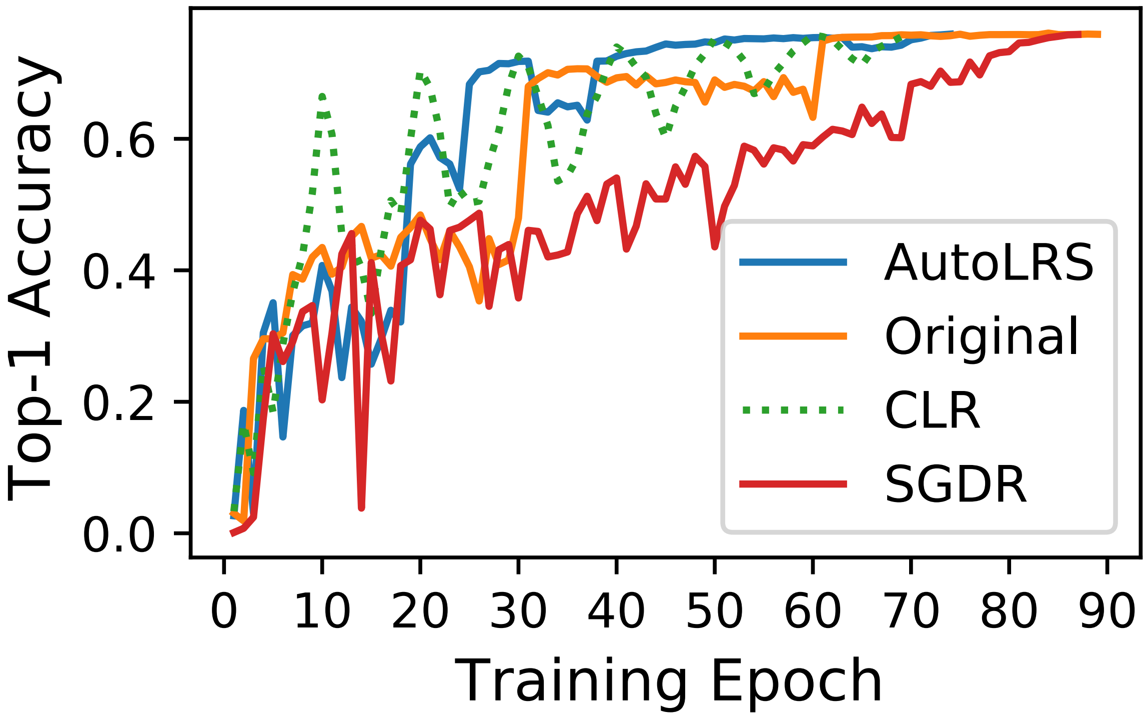
ResNet-50

1.22× faster convergence

LR schedules



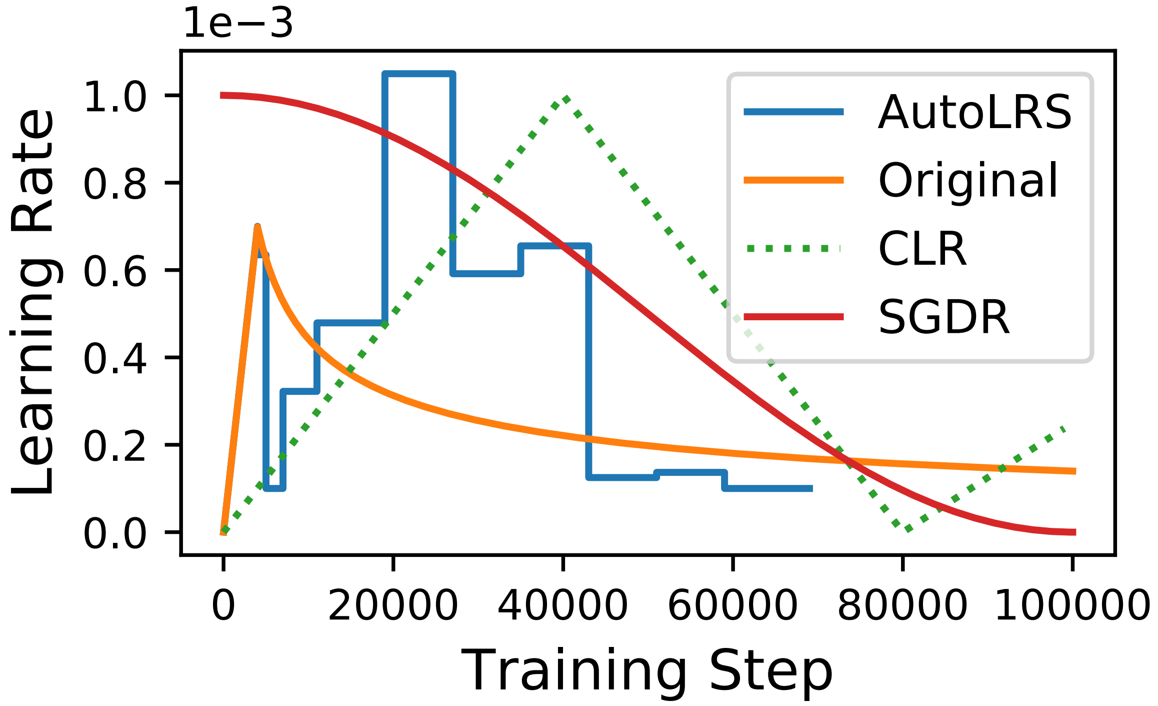
Top-1 Accuracy



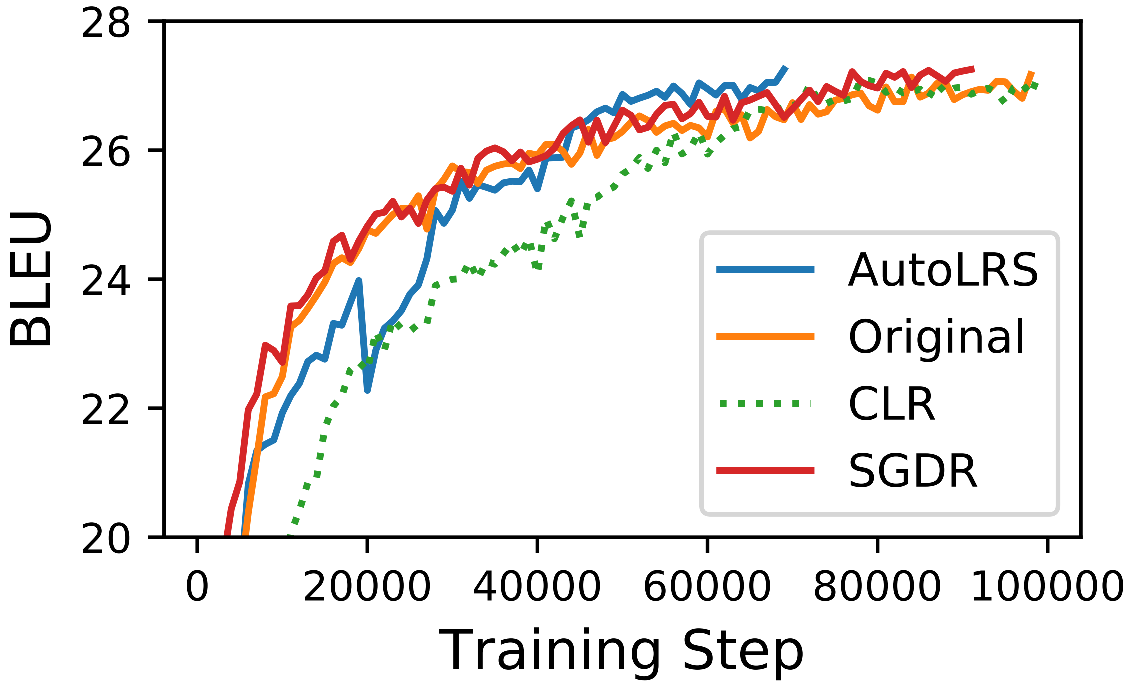
Transformer

1.43× faster convergence

LR schedules



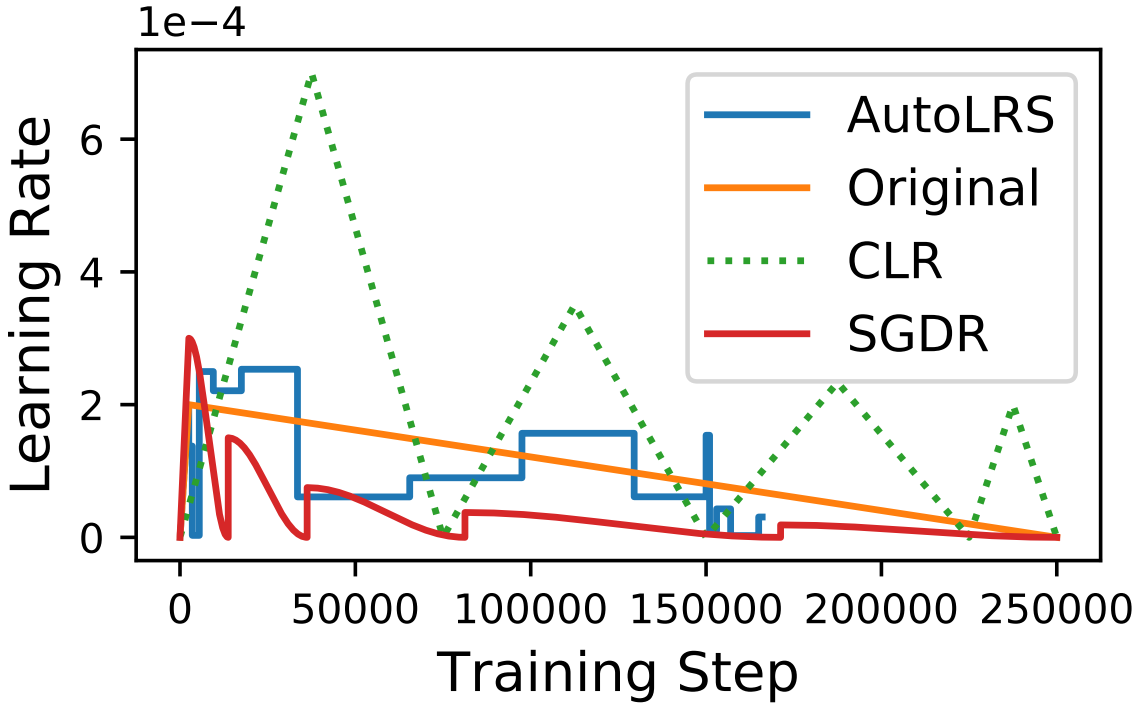
BLEU



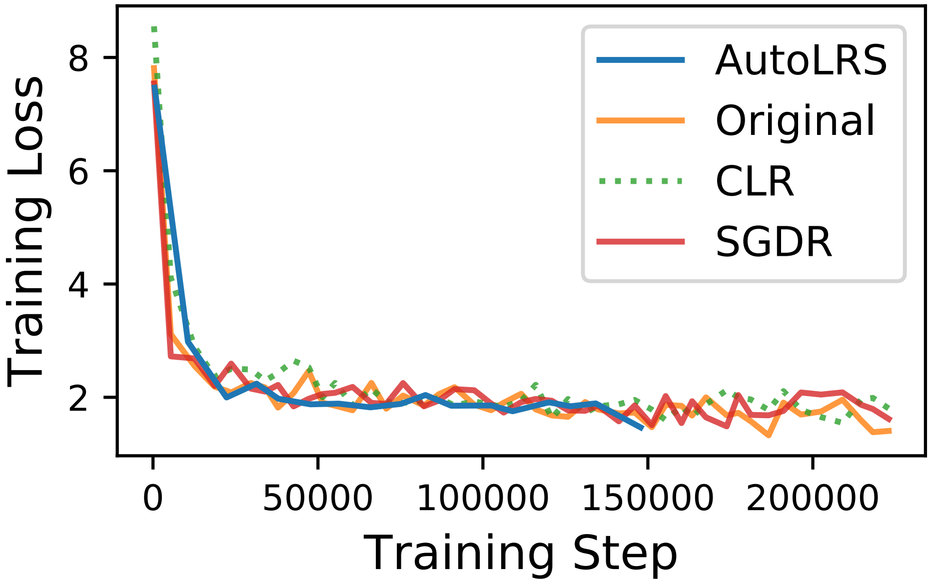
BERT

1.5× faster convergence

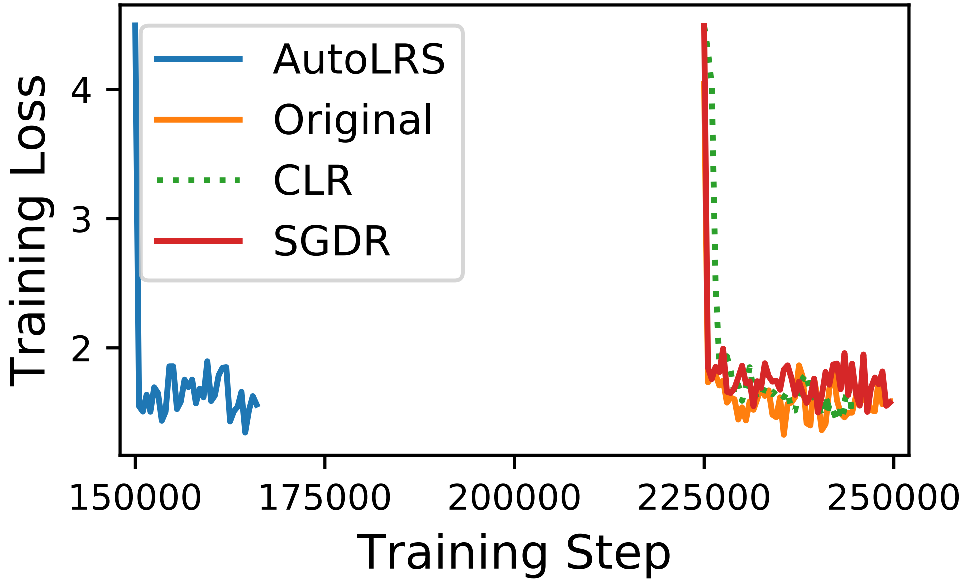
LR schedules (Phase 1 + 2)



Training loss in Phase 1



Training loss in Phase 2



AutoLRS Summary

➔ Aid ML practitioners with automatic and efficient LR schedule search for the DNNs

- We perform LR search for each training stage and solve it by Bayesian optimization.
- We train a light-weight exponential forecasting model from the training dynamics of BO exploration.
- *AutoLRS* achieves a speedup of 1.22x, 1.43x, and 1.5x on training ResNet-50, Transformer, and BERT compared to their highly hand-tuned LR schedules.

Give it a try: https://github.com/YuchenJin/autolrs